Jingwei+: A Distributed Large-Scale RDF Data Server

Xin Wang[1], Longxiang Jiang[2,*], Hong Shi[1], Zhiyong Feng[1], and Pufeng Du[1]

[1] School of Computer Science and Technology, Tianjin University, Tianjin, China
{wangx,serena,zyfeng,pdu}@tju.edu.cn
[2] School of Computer Software, Tianjin University, Tianjin, China
lxjiang2012@gmail.com

Abstract. With the development of the Linked Data project, the amount of RDF data published on the Web is steadily growing. To deal with the management issues of large-scale RDF data efficiently, we have developed a Bigtable-based distributed RDF repository, named Jingwei+, with a high level of scalability, to support the fast execution of triple pattern queries. The data service engine of Jingwei+ provides RESTful API to the outside, and its Web user interface have implemented the triple pattern query and the Linked Data navigational browsing. This paper demonstrates the query and navigation features of Jingwei+. The Jingwei+ RDF data server has laid the foundation of further development of the Semantic Web search engine.

Keywords: large-scale RDF data, distributed data server, storage scheme.

1 Introduction

RDF (Resource Description Framework) [1] has become the standard data model for representing and exchanging the machine-understandable information on the Semantic Web. Each Web resource is identified by an HTTP URI. RDF dataset is a finite set of triples (*S*, *P*, *O*), and each triple is expressed as three terms in which *S* occurs as subject, *P* as predicate and *O* as object. The subject, predicate and object can be represented as an HTTP URI, and the object can also be a literal or blank term. Fig. 1(a) shows an example of RDF triples.

With the development of the Linked Data project, more and more RDF data has been released on the Web. By September of 2011, the amount of triples on the Web has reached 31 billion [2]. Such large-scale RDF data and the inherent flexibility of RDF have posed new challenges to traditional data management systems. In order to manage these large-scale RDF data efficiently, we have developed a distributed RDF data server called Jingwei+. Firstly, Jingwei+ achieves the horizontal scalability and the high efficiency of RDF querying and processing by using the multi-dimensional nested key-value store. Secondly, the storage performance is promoted via namespaces transformation and compression. Thirdly, the user requests are fulfilled by the fast execution of triple pattern queries and the navigational browsing interface for the Linked Data. Fig. 2. shows the system architecture of Jingwei+.

* Corresponding author.

Q.Z. Sheng et al. (Eds.): APWeb 2012, LNCS 7235, pp. 779–783, 2012.

2 Features

This section presents the core features of Jingwei+ including RDF storage scheme based on Bigtable, namespace transformation and compression, and triple pattern queries.

2.1 RDF Storage Scheme Based on Bigtable

Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large size [3]. Our storage scheme is based on an extended Bigtable storage model which is a sparse, distributed and persistent storage scheme using the multi-dimensional sorted map. There are 5 levels of the mapping structure in our extended Bigtable storage scheme including *K*, *CF*, *SC*, *C* and *V* (underlined letters in Fig. 1(b)). Actually, we use *K* for storing *S*, *SC* for *P* and *C* for *O* as depicted in Fig. 1(b). To improve the query performance, the RDF triples are indexed in three different orders. For instance, *SPO* means we index RDF triples in the order of subject, predicate and object. There are three indexes: *SPO*, *POS* and *OSP*. We can get higher performance through trading space for time by using different indexes.

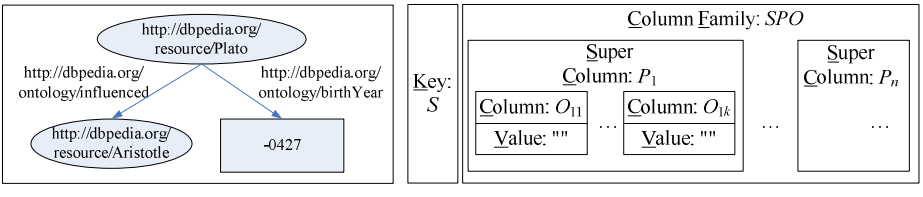

Fig. 1. (a) Example of RDF triples; (b) Extended Bigtable storage model for RDF

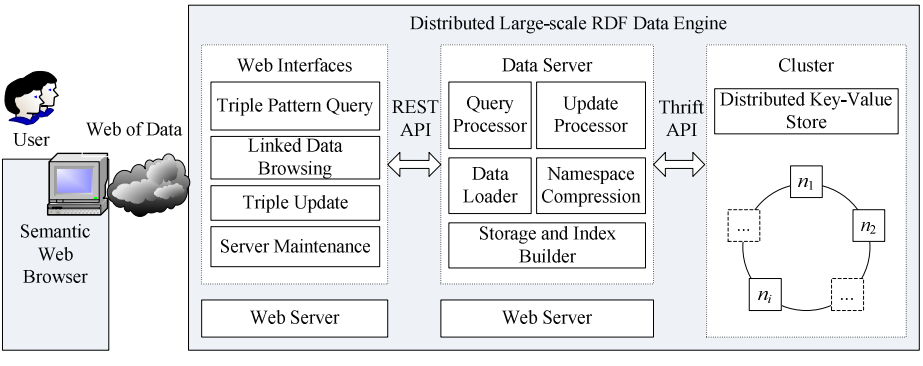

Fig. 2. System architecture of Jingwei+

2.2 Namespace Transformation and Compression

According to the RDF standard [1] and the principles of Linked Data [5], an RDF data set contains a large number of HTTP URI strings, and many of them share the same URI prefix. In order to reduce the storage space and improve the query performance, we transform high-frequency prefixes into simple abbreviations, thus compressing the URI strings.

2.3 Triple Pattern Query

A triple pattern query, which is the most fundamental component of an RDF query, is similar to an RDF triple except that it places the character "?" in front of a term to indicate a variable. In fact, triple pattern queries are essentially a subset of the SPARQL query language [4]. Therefore, triple pattern queries require an efficient processing mechanism. On the basis of our storage and indexing scheme, Jingwei+ selects the corresponding index for a query according to the variable occurrences.

3 Demonstration

This section presents the demonstration environment and scenarios of Jingwei+.

3.1 Demonstration Environment

Jingwei+ is written in Java, and the version of JDK is 1.6. There is an 8-nodes cluster in the underlying storage system. Each node has a 2.4 GHz CPU and 2 GB memory. The operating system is Ubuntu 10.04. We use Apache Tomcat 7.0 as the Web server. Currently, Jingwei+ has loaded two datasets: DBpedia [6] including ontology mapping based instances (13.8 million triples), short abstracts of DBpedia, images links and extended links to Geonames [7], and Geonames 2.2.1 RDF dump (145.9 million triples).

3.2 Demonstration Scenarios

In this section, we present the demonstration scenarios by the following use cases.

Use case 1: Triple pattern query. Input 7 different types of triple pattern queries in the Jingwei+ Web interface. Table 1 shows these query results and execution times. Fig. 3(a) shows the result of query t5.

Use case 2: Linked Data navigational browsing. On the basis of query t5, we can get more detail information if we click on the object Aristotle as shown in Fig. 3(a). Any resource can be displayed as long as a corresponding HTTP URI is available.

Use case 3: Query across datasets. Select a dataset before executing a triple pattern query. We can issue a query involving DBpedia URIs when the Geonames dataset is selected. Fig. 3(b) shows the results of querying Beijing (using the DBpedia URI) in Geonames. The default option is "All Datasets".

Table 1. Demonstration results for triple pattern query

Use case number	Triple pattern query	Number of results	Execution time / s
t1	(r:Plato, o:influenced, r:Aristotle)	1	0.005
t2	(r:Plato, o:mainInterest, *?X*)	10	0.007
t3	(r:Plato, *?X*, r:Aristotle)	1	0.008
t4	(*?X*, o:influencedBy, r:Plato)	79	0.011
t5	(r:Plato, *?X*, *?Y*)	52	0.006
t6	(*?X*, *?Y*, r:Plato)	88	0.017
t7	(*?X*, o:mainInterest, *?Y*)	2240	0.149

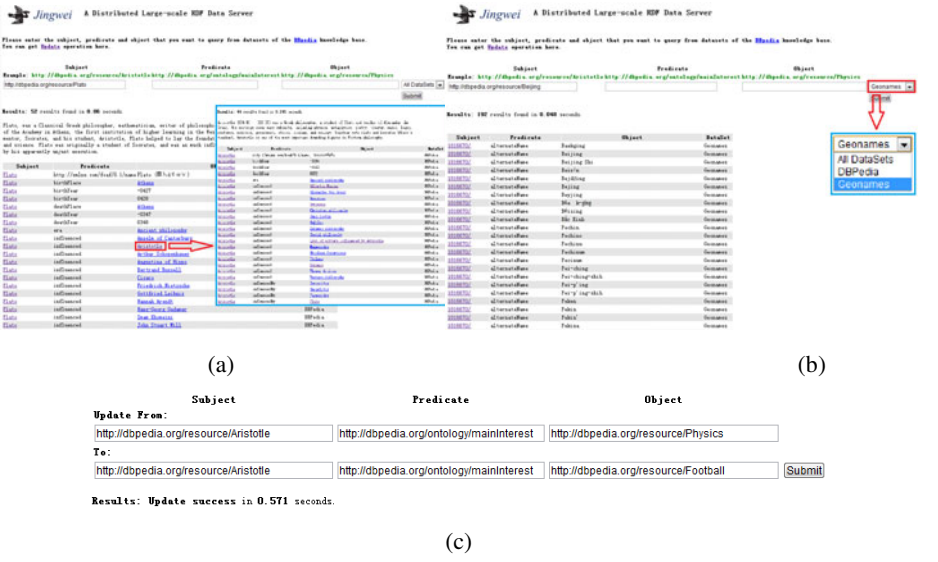

Fig. 3. (a) Triple pattern query and navigational browsing; (b) Query across datasets; (c) RDF triple update

Use case 4: RDF triple update. Input an original triple (r:Aristotle, o:mainInterest, r:Physics) and a new triple (r:Aristotle, o:mainInterest, r:Football). Then execute the update operation. Fig. 3(c) shows the screenshot of the update.

4 Conclusion

This paper describes the architecture of a novel distributed large-scale RDF data server called Jingwei+, and demonstrates its key features. In the future, many aspects of the system can be improved, such as the design of an inverted list for RDF keyword search and the analytical queries on large-scale RDF data using MapReduce. Also, we will conduct experiments to evaluate performances on Jingwei+ with other Linked Data datasets besides DBpedia and Geonames, and compare Jingwei+ with other related systems.

Acknowledgements. This work was supported by the National Science Foundation of China (No. 61100049, 61070202) and the Seed Foundation of Tianjin University (No. 60302022).

References

1. Klyne, G., Carroll, J.J., McBride, B.: Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, World Wide Web Consortium (2004)
2. Linked Data, `http://linkeddata.org/`
3. Chang, F., Dean, J., Ghemawat, S., Hsieh, C.W., Wallach, A.D., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: A Distributed Storage System for Structured Data. In: 7th USENIX Symposium on Operating Systems Design and Implementation, pp. 205–218. USENIX Association, Berkeley (2006)
4. Prud'Hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation, World Wide Web Consortium (2008)
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
6. DBpedia, `http://dbpedia.org/`
7. Geonames, `http://www.geonames.org/`